In an inelastic pipelined processor, instructions in the pipeline between the jump or branch and instructions fetched from the jump or branch target can be invalidated using an FF to indicate the valid state for the stage. When the control flow changes this FF can be set or cleared causing instructions in the pipeline to be ignored. This mechanism does not work however for an elastic pipeline where instructions are queued in fifos. Effectively every instruction stored in the pipeline would need to be tagged as invalid on a jump or branch, but instructions in a fifo are unaccessible. So, the concept of instruction streams is used.

Jumps and branches modify the stream of instructions viewed by the processing core. To track the instruction stream each instruction has a stream number associated with it. The current global stream number is assigned to the instruction when it is fetched. When a branch or jump occurs the global stream number is modified. At each pipeline stage the stream number is checked against the stream number that the stage is working on. If the stream number of the instruction does not match the stage’s stream number then the instruction is ignored. In a pipeline stage where the jump or branch takes place, the stream number for the stage is modified at the same time as the global stream number. Keeping these two consistent means instructions fetched along the new stream will be processed by the stage. One issue to be solved is that when the stream number is modified, stages after the one being modified also need their stream number modified for new instructions. So an indicator in the current instruction is passed to increment the stream number of the following stage.